

CLAIMS

1. A method of testing a process that downloads and installs customer ordered software onto a target computer, the method comprising:

- a. dynamically generating a file that contains instructions that when executed downloads and installs customer ordered software to a target computer;
- b. interpreting said dynamically generated file in accordance with a set of evaluation rules such that the outcome of the execution of said file is determined;
- c. analyzing the outcome of the execution of said file to determine possible syntax errors and possible flow errors;
- d. reporting said syntax errors and flow errors in a readable format.

2. A method as in claim 1 wherein said dynamically generated file is a main batch file created from a static text file that indicates the model type of the target computer, a lookup file that indicates the necessary instruction required to be executed for the model type indicated, and a process that reads the model type from said static text file and creates said dynamically generated file by reading said lookup file to determine command components.

3. A method as in claim 2 wherein said main batch file contains one or more labels identifying the flow of the process, one or more commands containing instructions to be executed and one or more calls to one or more static batch files.

4. A method as in claim 3 wherein the process of interpreting said dynamically generated batch file recursively simulates each of said one or more batch files to determine the outcome of the process.

5. A system of testing a process that downloads and installs customer ordered software onto a target computer, comprising:

- a. a simulation computer;
- b. a first process for creating a second process that downloads and installs customer ordered software onto a target computer;
- c. a third process for recursively simulating and interpreting the outcome of the execution of the second process;
- d. one or more output files that contain information relating to the simulation and interpretation of the second process.

6. A system as in claim 5 wherein said first process reads a electronic traveler to determine the model of the target computer, looks up in the master token list the model of the target computer and creates from the information in the master token list a second process that is an executable main batch file that downloads and installs customer ordered computer software onto the target computer;

7. A system as in claim 6 wherein said main batch file contains labels, commands and sub batch file calls, said third process interpretively tracks said labels, simulates each of said commands and recursively evaluates each of said sub batch files until the end of the main batch file is reached by said third process;

8. An automated method of testing the execution of a recursive batch file, said method comprising:

- a. analyzing the execution of the batch file;
- b. analyzing the execution of sub-batch files called by the batch file;
- c. reporting results of the analysis;

9. A method as in claim 8 wherein the batch file contains labels, commands and comments and the analysis comprises the steps of reading the batch file, creating a list of labels, opening the batch file and verifying the integrity of each label in the list in conjunction with the batch file.

10. A method as in claim 9 wherein analyzing the execution of the batch file further comprises reading each line in the batch file and applying rules to each line.

11. A method as in claim 8 wherein analyzing the execution of each of sub-batch files called by the batch file comprises the steps of reading the sub-batch file, creating a list of labels, opening the batch file and verifying the integrity of each label in the list in conjunction with the batch file.

12. A method as in claim 8 wherein analyzing the execution of the sub-batch files further comprises reading each line in the sub-batch files and applying rules to each line.